WIKIPEDIA

Gray-Code

Gray-Code	
stetig	ja
Hamming-Abstand	1

Der **Gray-Code** ist ein stetiger Code, bei dem sich benachbarte Codewörter nur in einer einzigen binären Ziffer unterscheiden, die Hamming-Distanz benachbarter Codewörter ist 1. Übertragungsfehler bei sich kontinuierlich ändernden digitalen Signalen auf mehradrigen Leitungen werden so verringert, da sich unterschiedliche Laufzeiten nicht auswirken können. Er dient als Kodierungsverfahren zur robusten Übertragung digitaler Größen über analoge Signalwege. Der Code ist nach dem Physiker Frank Gray benannt, welcher 1953 das Patent auf dieses Verfahren erhielt.[1]

Es kann ein balancierter Gray-Code konstruiert werden, wobei sich alle Stellen (bits) der Codewörter gleich häufig von benachbarten Codewörtern unterscheiden. Beim durchgehen aller Codewörter in korrekter Reihenfolge ändert sich dann jede Stelle gleich oft.[2]

Meistens ist der Gray-Code als Binärcode ausgeführt, kann aber auch für mehrstufige Übertragungswege benutzt werden.

Inhaltsverzeichnis

Generierung aus Binärcode
- Logische Operatoren
- Generatormatrix

Generierung als Gray-Zähler

Bedeutung
- Problem bei Dualcode-Signalen
- Lösung mit Gray-Code
- Karnaugh-Veitch-Diagramm

Geometrische und graphentheoretische Darstellung

Anwendungen

Beispiel

Einen Gray-Code zurückrechnen

Geschichte

Ähnliche Codes

Weblinks

Einzelnachweise

Generierung aus Binärcode

Logische Operatoren

Die folgenden Punkte zeigen, wie man Schritt für Schritt aus einem Binärcode eine Gray-codierte Binärzahl erhält:

- X1: Dualzahl im Binärcode
- X2: Rechts-Shift der Dualzahl um 1 Bit
- X3: Modulo-2-Addition (XOR-Verknüpfung) von X1 und X2; dies ist die gewünschte Zahl im Graycode.

Das gleiche als Pseudocode:

- Binärcode X1 → Graycode: X3 = (X1 XOR X2)

Und als Formel:

- $x_3 = x_1 \oplus (x_1 \gg 1)$

Generatormatrix

Da der Gray-Code ein linearer Code ist, kann man ihn mit einer Generatormatrix G erzeugen. Ein binäres Wort w der Länge n kann man als Vektor eines n-dimensionalen $\mathbb{F}_2$-Vektorraums $\mathbb{F}_2^n$ betrachten. Sei w nun ein Zeilenvektor, dann lässt sich die Kodierung des Wortes w in das Codewort c wie folgt darstellen:

$$c = w \underbrace{\begin{bmatrix} 1 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 1 \end{bmatrix}}_{=:G}$$

Die Dekodierung erfolgt mit der Multiplikation der Inversen G^{-1} von G. Diese hat folgende Form:

$$G^{-1} = \begin{bmatrix} 1 & \cdots & \cdots & 1 \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ 0 & & & 1 \end{bmatrix} \in \mathbb{F}_2^{n \times n}$$

Der Vektorraum $\mathbb{F}_2^n$ lässt sich anschaulich mit Hyperwürfeln darstellen.

Generierung als Gray-Zähler

Man kann auch direkt einen Gray-Code-Zähler in Hardware (z. B. in HDL) programmieren. Hierzu ist es hilfreich, ein Hilfsregister zu benutzen, das mit jedem Taktzyklus toggelt.

```
Qh [n+1] = !Qh [n]
Qh [0  ] =  0     wenn der Gray-Code-Zähler mit Null startet also Q[0]=0, oder Q[0] eine gerade Anzahl an Einsen hat. Bei anderer Initialisierung
würde der Zähler rückwärts laufen.
```

Damit wird die Kombinatorik recht übersichtlich:

```
Q0  [n+1] = !(Q0  [n] ^   Qh  [n] )
Q1  [n+1] =   Q1  [n] ^ ( Q0  [n] &  Qh  [n] )
Q2  [n+1] =   Q2  [n] ^ ( Q1  [n] & !Q0  [n] &  Qh [n] )
Q3  [n+1] =   Q3  [n] ^ ( Q2  [n] & !Q1  [n] & !Q0 [n] & Qh [n] )
          …
Qk-1[n+1] =   Qk-1[n] ^ ( Qk-2[n] & !Qk-3[n] & …  & !Q1 [n] & !Q0 [n] & Qh [n])
Qk  [n+1] =   Qk  [n] ^ (!Qk-2[n] &            …  & !Q1 [n] & !Q0 [n] & Qh [n])
```

Der Unterschied zwischen den Formel für das größte Bit Q_k und den kleineren Bits Q_{k-1} ist nötig, damit der Zähler zyklisch ist, also der Zähler nach dem letzten Wert Q=100…00 auf den Anfangswert Q=000…00 springt. Bei einem unendlichen Zähler gäbe es keinen Unterschied.

```
^ := Exklusiv Oder / XOR / Antivalenz

! := Inverter / NOT / Negation

& := Und / AND / Konjunktion
```

Bedeutung

Motivation für die Entwicklung dieses Codes ist das folgende Problem: Auf mehreren Adern einer elektrischen Datenleitung sollen Daten parallel übertragen werden, die sich stetig (also immer nur um ein Digit) ändern, typisch dafür sind z. B. Signale eines Temperatursensors oder eines Drehwinkelgebers. Als Dualzahl übertragen, ändern sich die Bits bei einem neuen Messwert auf jeder betroffenen Leitung theoretisch exakt gleichzeitig, und zwar sowohl am Eingang der Leitung als auch am Ausgang. Tatsächlich aber ändern sich die Bits auf der Leitung nicht gleichzeitig. Das kann verschiedene Ursachen haben: Bauteilestreuung, Laufzeiten, Asymmetrien usw. Dadurch kommt es zu ungewollten Zwischenzuständen und kurzzeitig (zwischen den roten Linien) falsch empfangenen Werten:

Problem bei Dualcode-Signalen

2-Bit-Gray-Code:

```
00
01
11
10
```

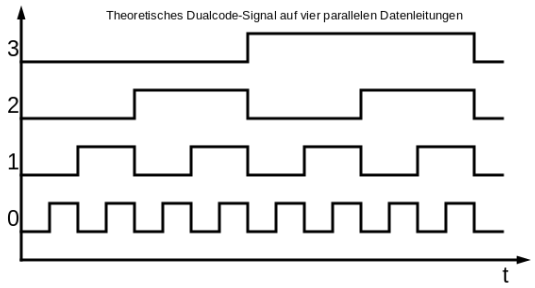
3-Bit-Gray-Code:

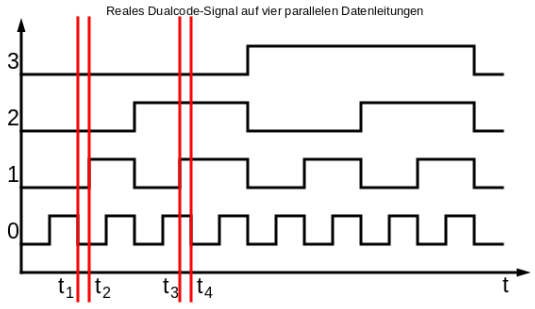
```
000
001
011
010
110
111
101
100
```

4-Bit-Gray-Code:

6-Bit-Gray-Code:

```
000000
000001
000011
000010
000110
000111
000101
000100
001100
001101
001111
001110
001010
001011
001001
001000
011000
011001
011011
011010
011110
011111
011101
011100
```






```
0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000
```

5-Bit-Gray-Code:

```
00000
00001
00011
00010
00110
00111
00101
00100
01100
01101
01111
01110
01010
01011
01001
01000
11000
11001
11011
11010
11110
11111
11101
11100
10100
10101
10111
10110
10010
10011
10001
10000
```

```
010100
010101
010111
010110
010010
010011
010001
010000
110000
110001
110011
110010
110110
110111
110101
110100
111100
111101
111111
111110
111010
111011
111001
111000
101000
101001
101011
101010
101110
101111
101101
101100
100100
100101
100111
100110
100010
100011
100001
100000
```

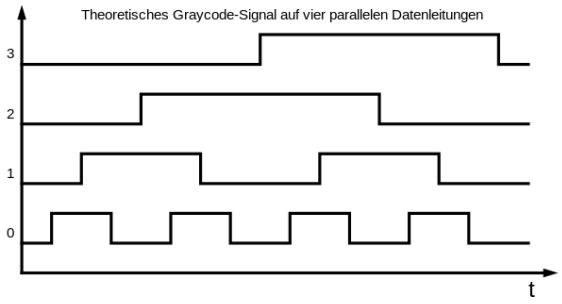
Während das theoretische Signal in der Reihenfolge

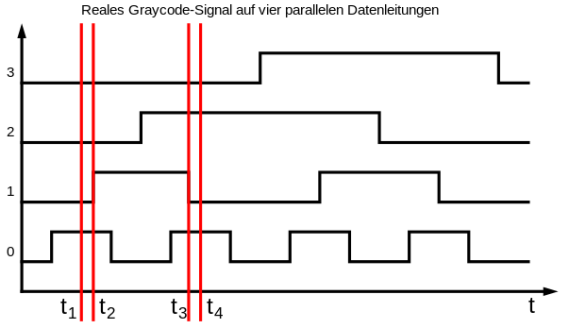
- {0000}, {0001}, {0010}, {0011}, {0100}, {0101}, {0110}, {0111} usw.

abgesendet wird, kommen am Ausgang kurzzeitig andere Signalzustände an:

- {0000}, {0001}, **{0000}**, {0010}, {0011}, {0100}, {0101}, **{0111}**, {0110}, {0111} usw.

Lösung mit Gray-Code





Um das zu vermeiden, werden die Steuersignalzustände mittels Gray-Code abgesendet, sodass sich immer nur ein Bit gleichzeitig ändert:

- Abgesendete Sequenz: {0000}, {0001}, {0011}, {0010}, {0110}, {0111}, {0101}, {0100} usw.
- Ankommende Sequenz: {0000}, {0001}, {0011}, {0010}, {0110}, {0111}, {0101}, {0100} usw.

Hier kommt also am Ausgang auch dann die gleiche Sequenz wie am Eingang an, wenn beachtliche Zeitfehler (rote Linien) auftreten.

Karnaugh-Veitch-Diagramm

Im Karnaugh-Veitch-Diagramm erkennt man den Graycode – es sind mehrere Sequenzen möglich – daran, dass Übergänge nur zwischen (horizontal

oder vertikal) benachbarten Feldern vorkommen.

Reihenfolge Dualcode

	$\neg X_0$	X_0	X_0	$\neg X_0$	
$\neg X_2$	0	1	3	2	$\neg X_3$
X_2	4	5	7	6	$\neg X_3$
X_2	12	13	15	14	X_3
$\neg X_2$	8	9	11	10	X_3
	$\neg X_1$	$\neg X_1$	X_1	X_1	

Reihenfolge Graycode

	$\neg X_0$	X_0	X_0	$\neg X_0$	
$\neg X_2$	0	1	2	3	$\neg X_3$
X_2	7	6	5	4	$\neg X_3$
X_2	8	9	10	11	X_3
$\neg X_2$	15	14	13	12	X_3
	$\neg X_1$	$\neg X_1$	X_1	X_1	

Der Code eignet sich auch für zyklische Anwendungen wie der unten abgebildeten Scheibe, da sich auch beim Übergang von der höchsten Zahl auf die Null nur eine Stelle ändert.

Die Wertigkeit einer 1 an der Position n im Gray-Code Zahlensystem ist $2^n - 1$ (wobei n ab 1 zählt, also … 31, 15, 7, 3, 1). Die einzelnen Einsen werden, im Gegensatz zum normalen Binärsystem, nicht addiert, sondern von rechts beginnend subtrahiert. Beispiel: 111_{Gray} = 7 - (3 - 1) = 5 oder 1111_{Gray} = 15- (7 - (3 - 1)) = 10. Stellen, die 0 sind, werden dabei ausgelassen, Beispiel: 101_{Gray} = 7 - 1 = 6.

Bei der Generierung von Gray-Code wird symmetrisch vorgegangen.

Da sich benachbarte Werte nur in einer Ziffer unterscheiden, ist der Gray-Code geeignet, um Fehler in seriellen Prozessen aufzudecken.

Geometrische und graphentheoretische Darstellung

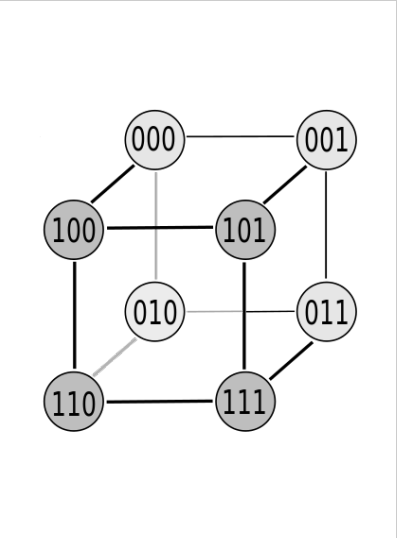


Bild 1: Würfel

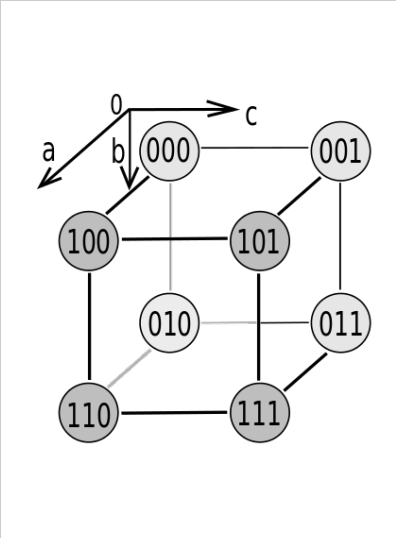


Bild 2: Würfel mit Koordinatensystem

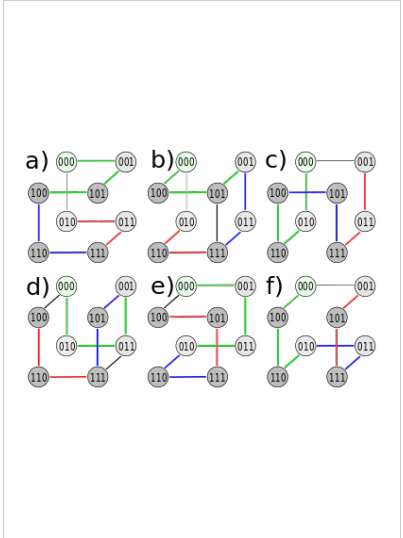


Bild 3: Die 6 Pfade zu dem Gray-Code in der Tabelle. Es handelt sich um einen Hamiltonkreis. Startpunkt: 000 (grüner Kreis jeweils links oben), Fortsetzung: grüne→blaue→rote→schwarze Linie, Endpunkt: am Startpunkt

Bild 1 zeigt den Hexaeder für 3 Variablen und Bild 2 den gleichen Würfel mit dazugehörigem Koordinatensystem. Die Knoten (Eckpunkt oder Kreise) am Einheitswürfel entsprechen jeweils einer Zeile im Gray-Code. Die Übergänge (Nachbarschaft der Zeilen) sind durch die Kanten des Würfels symbolisiert. Beim Wandern auf der Kante entsteht ein Gray-Code.

geschlossener 3-Bit-Gray-Code

a)	b)	c)	d)	e)	f)
000	000	000	000	**000**	000
001	100	010	010	**001**	100
101	101	110	011	**011**	110
100	001	100	001	**010**	010
110	011	101	101	**110**	011
111	111	111	111	**111**	111
011	110	011	110	**101**	101
010	010	001	100	**100**	001

Auf jeder Kante ändert sich genau 1 Bit. Der Gray-Code hat so viel Nachbarschaften, wie der Würfel Kanten hat. Aus dem Würfel in Bild 1 können die möglichen Pfade auf 6 verschiedenen Wegen durchschritten werden. Somit ergeben sich 6 Möglichkeiten, um einen 3-Bit-Gray-Code zu erzeugen, der die Bedingungen des Gray-Codes erfüllt (Tabelle und Bild 3). Abgesehen davon ist der Gray-Code zyklisch und der Startpunkt könnte deshalb auch an einer anderen Zeile sein. Wegen seiner einfachen rekursiven Generierungsvorschrift wird meist der binäre reflektierte Gray-Code (binary-reflected Gray code) angegeben (Spalte „e“ – vorletzte Spalte in der Tabelle). Es gibt für eine bestimmte Bitlänge eine ganze Klasse von Graycodes.

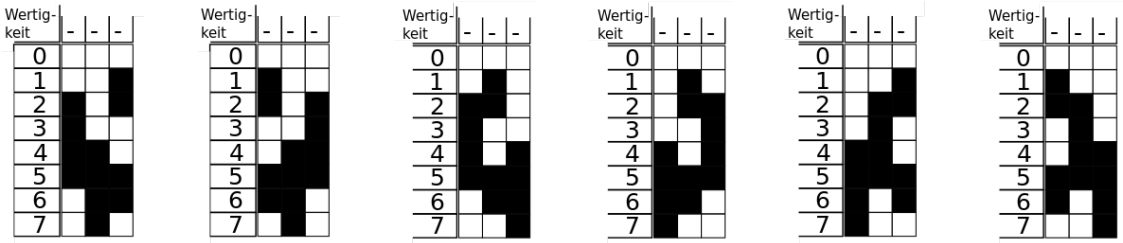
Es gibt für einen n-Bit-Gray-Code exakt so viel Varianten, wie es Hamiltonkreise auf einem n-dimensionalen Hyperwürfel gibt.

geschlossener 3-Bit-Gray-Code

a)	b)	c)	d)	e)	f)
000	000	010	010	000	100
001	100	110	011	001	110
101	101	100	001	011	010
100	001	101	101	010	011
110	011	111	111	110	111
111	111	011	110	111	101
011	110	001	100	101	001
010	010	**000**	**000**	100	**000**

Da der hier dargestellte Gray-Code zyklisch ist, wurde in dieser Tabelle der Code in den Spalten c), d) und f) um eine Stelle nach oben verschoben (im Vergleich zur Tabelle weiter oben), so dass jeweils die drei Nullen in der letzten Tabellenzeile stehen. So ist erkennbar, dass es sich bei dem Gray-Code in Spalte a) nur um eine spiegelbildliche Umkehrung der Spalte b) handelt. Genauso ist Spalte c) die Umkehrung von Spalte d), während Spalte e) die Umkehrung von Spalte f) ist. Es gibt drei ungerichtete Hamiltonkreise am dreidimensionalen Würfel, die hier lediglich in unterschiedlicher Richtung (gerichteter Hamiltonkreis) dargestellt wurden.

Zur besseren Veranschaulichung sind hier nochmals die Codetafeln für die 6 Varianten des 3-Bit-Graycodes dargestellt. Wobei die Variante e den binären reflektierten Gray-Code darstellt, der meist gemeint ist, wenn vom Gray-Code die Rede ist. Die 6 Versionen kann man auch durch Permutation der 3 Spalten der Codetafel erzeugen. Daraus ergibt sich, dass es bei n Bit n! Versionen gibt. Also für 3 Bit 3!= 6 Versionen des 3-Bit-Graycodes.



3 Bit: a 3 Bit: b 3 Bit: c 3 Bit: d 3 Bit: e 3 Bit: f

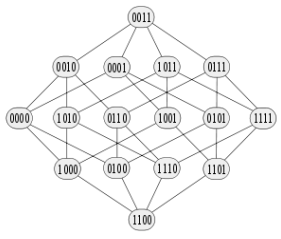
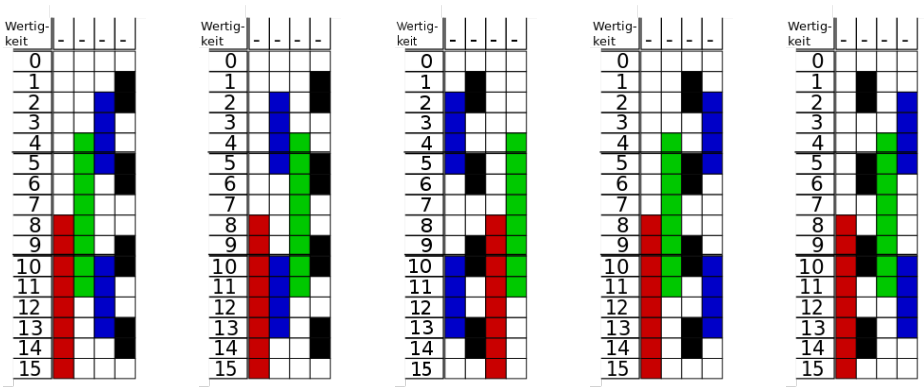


Bild 4

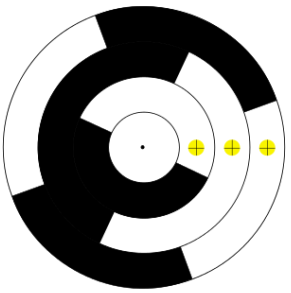
…
usw.
…



Den 4-Bit-Gray-Code kann man aus dem Hyperwürfel in Bild 4 ablesen. Für 4 Bit gibt es 4! = 24 verschiedene Gray-Codes.

Anwendungen

Eine Anwendungsmöglichkeit ist die Bestimmung der absoluten Position einer Scheibe oder Leiste, die mit schwarzen und weißen Balken markiert ist, die mit Lichtschranken oder anderen Sensoren abgetastet werden. Diese Position wird dann zur Winkel- oder Drehgeschwindigkeitsmessung verwendet.



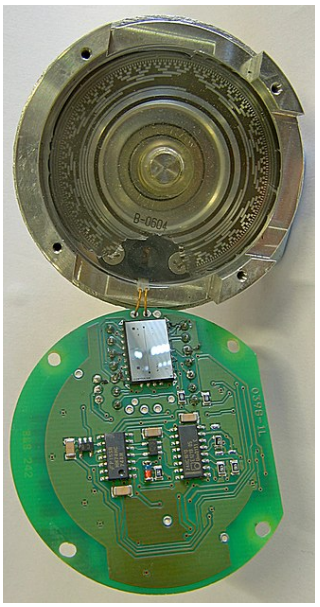
Schemazeichnung einer Scheibe mit Gray-Codierung. Die gelben Punkte stellen Lichtsensoren dar.

Eine weitere Anwendung ist die Streifenprojektion. Dort wird eine Folge von Mustern aus parallelen Streifen auf ein Objekt projiziert. Die Nummer der Streifen ist Gray-kodiert und kann von einer beobachtenden Kamera für jeden Bildpunkt berechnet werden.

Eine andere Anwendung ist das asynchrone Einlesen von Daten. Beispielsweise wird der Gray-Code genutzt, um in Korrelatoren die Zählerstände fehlerfrei einzulesen. Selbst im ungünstigsten Fall, wenn während eines kippenden Bits eingelesen wird, ist das Ergebnis immer korrekt, da ein kippendes Bit nicht definiert ist und es zudem nur einen Unterschied von ±1 ausmacht. Diese Art des Einlesens erfordert keine Synchronisation und nur sehr wenig CPU-Zeit.

Weitere Anwendungsmöglichkeiten sind Windrichtungsmesser oder Wasserniveaumesser, Abbildung des Fahrkorbstands bei Aufzügen.

Der reflektierte Gray-Code hat eine enge Beziehung zur Lösung des Problems der Türme von Hanoi, und er beschreibt auch den Lösungsweg der Chinesischen Ringe.



Ein Gray-Code Absolutwertgeber mit 13 bits

Beispiel

Die Dezimalzahl $4_{10} = 100_2$ entspricht dem Gray-Code $6_{10} = 110_2$. Die Dekodierung in die Dezimaldarstellung folgt dann der Regel $1 \cdot 7_{10} - (1 \cdot 3_{10} - 0 \cdot 1_{10}) = 4_{10}$. Wenn mehrere Einsen in einer Gray-Code-Zahl vorkommen, werden diese voneinander subtrahiert: Der Gray-Code $7_{10} = 111_2$ wird wie folgt dekodiert: $1 \cdot 7_{10} - (1 \cdot 3_{10} - 1 \cdot 1_{10}) = 5_{10}$.

Allgemeines Verfahren: Bei einer Umwandlung ist entscheidend, an welcher Position die Einser stehen. Die Position hat Einfluss auf die Rechnung. Wenn wir uns die Zahl 100 anschauen, dann steht die Eins auf Position 3 (von rechts nach links). Den Faktor für die Eins bekommt man, indem man sich überlegt, welche Dezimalzahl maximal in einer 3-Bit Zahl binär gespeichert werden kann. In 3 Bit Binärcode kann maximal die Zahl 7 (111) gespeichert werden. Nehmen wir jetzt eine größere Binärzahl, funktioniert das praktisch analog. Binärzahl: 11010 (1 an Position 5,4 und 2). 5 Bit Binärzahl: max. 31 4 Bit Binärzahl: max. 15 2 Bit Binärzahl: max. 3

Berechnung: $31_{10} - (15_{10} - 3_{10}) = 19$

Einen Gray-Code zurückrechnen

```
   for I := NumBits - 1 downto 0 do        // jedes einzelne Bit vom letzten bis zum ersten
     Value := Value or (                   // das Ergebnis jedes errechneten Bits dem Gesamtergebnis
hinzufügen
       (((1 shl (I + 1)) and Value) shr 1) // das Bit der Stelle zuvor im Ergebnis
       xor                                 // xor mit
       ((1 shl I) and GrayCode)            // der aktuellen Stelle des Codes
                     );
```

Geschichte

Noch bevor die Bezeichnung Gray-Code eingeführt wurde, gab es bereits mathematische Knobelspiele, in denen das Prinzip angewendet wurde. Erst später fand der Code die Beachtung von Ingenieuren. Bereits 1874 wendete Otto Schäffler, der in Wien Telegrafen und Telefone produzierte und verbesserte, den reflektierten Gray-Code an. Der Franzose Jean-Maurice-Émile Baudot verwendete Gray-Codes im Jahr 1874 für die elektrische Telegrafie. Er erhielt für seine Arbeit die Auszeichnung der französischen Ehrenlegion.

Namensgebend war allerdings Frank Gray, Forscher in den Bell Laboratories, der den schon 1941 von George Stibitz beschriebenen Code 1947 für seine Zwecke wiederentdeckte. Unter dem Titel *Pulse Code Communications* wurde am 17. März 1953 unter der US-Patentnummer 2,632,058[3] ein Patent für eine Gray-kodierende Elektronenröhre erteilt.

Ähnliche Codes

- 1-aus-10-Code
- Aiken-Code
- BCD-Code
- Gillham-Code
- Libaw-Craig-Code
- Stibitz-Code

Weblinks

Commons: Gray code (https://commons.wikimedia.org/wiki/Category:Gray_code?uselang=de) – Sammlung von Bildern, Videos und Audiodateien

- Java-Beispiel für die Umwandlung einer Dezimalzahl in Gray-Code Darstellung (https://web.archive.org/web/20070930040812/http://etc.manuel-breitfeld.de/gray-code-java-implementierung.html) (Memento vom 30. September 2007 im *Internet Archive*)
- Algorithmen zur Erzeugung des reflektierten Gray-Codes (http://combos.org/bits) auf dem Combinatorial Object Server (http://combos.org)

Einzelnachweise

1. Patent US2632058 (https://worldwide.espacenet.com/publicationDetails/biblio?locale=de_EP&CC=US&NR=2632058) : *Pulse code communication.* Veröffentlicht am 17. März 1953, Erfinder: Frank Gray.
2. Girish S. Bhat, Carla D. Savage: *Balanced Gray Codes*. In: *The Electronic Journal of Combinatorics*. 28. August 1996, ISSN 1077-8926 (https://zdb-katalog.de/list.xhtml?t=iss%3D%221077-8926%22&key=cql), S. R25–R25, doi:10.37236/1249 (https://doi.org/10.37236/1249) (combinatorics.org (https://www.combinatorics.org/ojs/index.php/eljc/article/view/v3i1r25) [abgerufen am 22. Mai 2021]).
3. US-Patentnummer 2,632,058 (http://www.freepatentsonline.com/2632058.html) freepatentsonline.com

Abgerufen von „https://de.wikipedia.org/w/index.php?title=Gray-Code&oldid=214890628“

Diese Seite wurde zuletzt am 19. August 2021 um 17:21 Uhr bearbeitet.

Der Text ist unter der Lizenz „Creative Commons Attribution/Share Alike“ verfügbar; Informationen zu den Urhebern und zum Lizenzstatus eingebundener Mediendateien (etwa Bilder oder Videos) können im Regelfall durch Anklicken dieser abgerufen werden. Möglicherweise unterliegen die Inhalte jeweils zusätzlichen Bedingungen. Durch die Nutzung dieser Website erklären Sie sich mit den Nutzungsbedingungen und der Datenschutzrichtlinie einverstanden.
Wikipedia® ist eine eingetragene Marke der Wikimedia Foundation Inc.